Journal of AI and Data Mining
Vol 4, No 1, 2016, 1-8
10.5829/idosi.JAIDM.2016.04.01.01

An improved joint model: POS tagging and dependency parsing

A. Pakzad and B. Minaei Bidgoli*

Department of Computer Engineering, Iran University of Science & Technology, Tehran, Iran.

Received 25 September 2015; Accepted 12 October 2015

*Corresponding author: b_minaei@iust.ac.ir (B. Minaei).

Abstract

Dependency parsing is a way of syntactic parsing and a natural language that automatically analyzes the dependency structure of sentences, and the input for each sentence creates a dependency graph. Part-Of-Speech (POS) tagging is a prerequisite for dependency parsing. Generally, dependency parsers do the POS tagging task along with dependency parsing in a pipeline mode. Unfortunately, in pipeline models, a tagging error propagates, but the model is not able to apply useful syntactic information. The goal of joint models simultaneously reduce errors of POS tagging and dependency parsing tasks. In this research, we attempted to utilize the joint model on the Persian and English language using Corbit software. We optimized the model's features and improved its accuracy concurrently. Corbit software is an implementation of a transition-based approach for word segmentation, POS tagging and dependency parsing. In this research, the joint accuracy of POS tagging and dependency parsing over the test data on Persian, reached 85.59% for coarse-grained and 84.24% for fine-grained POS. Also, we attained 76.01% for coarse-grained and 74.34% for fine-grained POS on English.

Keywords: *Joint model, Part-Of-Speech, Dependency Parsing, Persian Language.*

1. Introduction

POS tagging and dependency parsing are two important tasks in natural language processing. POS tagging is a preliminary step in the dependency-parsing task. An incorrect POS tag propagates errors in dependency parsing, but POS tagging is unable to use syntactic information.

A POS tagging and dependency parsing system for the Persian language suffers from error propagation, but it cannot use syntactic information for POS tagging. Hatori et al. (2012) presented an incremental joint model for POS tagging and dependency parsing on the Chinese language using Corbit software [1]. However, in this research, we reconciled the joint model of the Chinese language to the Persian language; the model's features were also optimized for Persian and English. Further, the joint accuracy for POS tagging and unlabeled dependency parsing for coarse-grained POS and fine-grained POS on Persian were 85.59% and 84.24%, respectively. Also, we reached 76.01% for coarse-grained and 74.34% for fine-grained POS on English. Experimental results on the Persian Syntactic Dependency Treebank1.0 and Universal Dependencies English Web Treebank v1.0 showed that our improved joint model significantly improved both POS tagging and dependency parsing accuracies compared to the pipeline model.

2. Related work

Bohnet and Nivre (2012) proposes a transition-based model for joint POS tagging and labeled dependency parsing with non-projective trees on the Chinese language [2]. This joint model uses beam search inference and global structure learning. Globally learned models can use richer feature space than locally trained models.

Hatori et al. (2011) presents the first incremental approach to the task of joint POS tagging and dependency parsing on Chinese [3]. We used this method in our research. In this approach, given a segmented sentence the model simultaneously considers POS tags and dependency relations within the given beam, and outputs the best parse along with POS tags. This incremental joint model has two problems: First, since the combined search space is huge, efficient decoding is difficult and naïve use of the beam is probable contributing to a decline in the search quality. Second, since the suggested model performs joint POS tagging and dependency parsing from left to right of the sentence, the model cannot exploit look-ahead POS tags to decide the next action. To deal with a huge search space, the model uses a dynamic programming (DP) extension for shift-reduce parsing, which allows the model to merge equal parser states and increases speed and accuracy. The model solves the lack of look-ahead POS information problem by delayed features. The delayed features include undetermined POS tags which are evaluated when the look-ahead POS tags are specified. This joint model is language-independent. Li et al. (2011) proposes graph-based joint models according to syntactic features. It defines first-, second-, and third-order joint models [4].

Li et al. (2012) presents a graph-based joint model. The POS tagging task does not profit much from a joint model because on average the POS features score is only 1/50 of the syntactic features in the joint results [5]. In other words, the POS features do not have much effect on determining the best joint result. The proposed model separately updates the POS features weights and the syntactic feature weights, and increases the weights of POS features in the joint optimization framework. This model improves POS tagging and dependency parsing accuracies.

Being available on the Persian language, first the data has been tagged and then has been used for dependency parsing. Seraji et al. (2012) presents two dependency parsers for the Persian language [6]. MaltParser and MSTParser are transition-based and graph-based dependency parsers, respectively. Both parsers are trained on the Uppsala Persian Dependency Treebank. The unlabeled attachment score for MaltParser and MSTParser are 74.81% and 71.08%, respectively. Those results are not comparable with our joint model results because the dataset is different. Khallash et al. (2013) studies the effect of morphological and lexical features on dependency parsing for Persian [7]. It studies the effect of features on the transition-based dependency parser MaltParser and the graph-based dependency parser MSTParser. Labeled attachment score with gold POS tags for MaltParser and MSTParser are 86.98% and 86.81%, respectively. Unlabeled attachment scores are not reported.

3. Baseline models

First, we introduce both a baseline POS tagger and a dependency parser. A combination of baseline models make-up the pipeline model. Then we describe the joint model and its default features. Added and subtracted features of the model and the logic behind each are discussed in the section 4.1.2. The dataset is divided into train, validation and test sets. Train and validation sets are used to determine the model's parameters for intermediate experiments, train and test sets are used for the final experiments. Corbit software is an unlabeled dependency parser. All of accuracies, which are reported in this article, are unlabeled attachment scores. Corbit reports POS tagging and dependency parsing accuracies with DEP and POS, respectively. We added a new DepPos accuracy measure, which shows the correctness of the POS tag and the dependency relation and the word's head simultaneously.

3.1. Baseline POS tagger

The Stanford Log-linear Part-Of-Speech Tagger was used in this research. This software is an implementation of Log-Linear POS Taggers with java as described in [8].

3.2. Baseline dependency parser

Corbit software has several different run modes [1]:

1- SegTag: Joint segmentation and POS tagging model.
2- Dep’: dependency parser.
3- Dep: Dep’ without look-ahead features.
4- TagDep: joint POS tagger and dependency parser .
5- SegTag+Dep/SegTag+ Dep’: a pipeline combination of SegTag and Dep or Dep’.
6- SegTagDep: joint segmentation and POS tagging and dependency parsing model.

In this research, we used the Dep’ mode which uses the shift-reduce parsing method as a baseline dependency parser.

3.3. Pipeline POS tagging and dependency parsing model

First, the data was tagged, and then we used the tagged data for dependency parsing with baseline dependency parsing.

4. Joint POS tagging and dependency parsing model

In this research, we used a joint POS tagging and dependency parsing model proposed by Jun Hatori [1] as a base model. We reconciled the model for Persian and English, and then we optimized the model's features.

4.1. Features

4.1.1. Default model's features

The joint POS tagging and dependency parsing model uses baseline dependency parser features represented in figure 1. In addition to these features, it uses syntactic features for POS tagging and delayed features. Figures 2(a) and 2(b) show the delayed features and syntactic features lists of a joint model.

$$
\begin{array}{l}
w_j \quad t_{j-1} \quad t_{j-1} \circ t_{j-2} \quad w_{j+1}{}^{1)} \\
w_j \circ E(w_{j-1})^{2)} \qquad w_j \circ E(w_{j+1})^{2)} \\
E(w_j) \circ w_j \circ E(w_{j+1})^{3)} \\
B(w_j) \quad E(w_j) \quad P(B(w_j)) \quad P(E(w_j)) \\
C_n(w_j) \quad (n \in \{2.\ldots.len(w_j) - 1\}) \\
B(w_j) \circ C_n(w_j) \quad (n \in \{2.\ldots.len(w_j)\}) \\
E(w_j) \circ C_n(w_j) \quad (n \in \{2.\ldots.len(w_j) - 1\}) \\
C_n(w_j) \quad (if\ C_n(w_j) equals\ to\ C_{n+1}(w_j)) \\
\hline
1)\ if\ len(w_{j+1}) < 3; \quad 2) if\ len(w_j) < 3; \\
\qquad 3)\ if\ len(w_j) = 1
\end{array}
$$

Figure 1. Feature templates for baseline POS tagger, where t_i is the tag assigned to the i-th word w_i, B(w) and E(w) is the beginning and the ending character of word w, $C_n(w)$ is the n-th character of w, P(c) is the set of tags associated with the single-character word c based on the dictionary [3].

(a)	$q_0.t \quad q_0.w \circ q_0.t$	$s_0.t \circ q_0.t$
	$s_0.t \circ q_0.t \circ q_1.t$	$s_1.t \circ s_0.t \circ q_0.t$
	$s_0.w \circ q_0.t \circ q_1.t$	$s_1.t \circ s_0.w \circ q_0.t$
(b)	$t \circ s_0.w$	$t \circ s_0.t$
	$t \circ s_0.w \circ q_0.w$	$t \circ s_0.t \circ q_0.w$
	$t \circ B(s_0.w) \circ q_0.w$	$t \circ E(s_0.w) \circ q_0.w$
	$t \circ s_0.t \circ s_0.rc.t$	$t \circ s_0.t \circ s_0.lc.t$
	$t \circ s_0.w \circ s_0.t \circ s_0.rc.t$	$t \circ s_0.w \circ s_0.t \circ s_0.lc.t$

Figure 2. (a) List of delayed features for joint parser; (b) Syntactic features for the joint parser, where t is the POS tag to be assigned to q0 [3].

4.1.2. Features optimization

POS's are classifications of words based on their functions in sentences for purposes of grammatical analysis. Each coarse-grained POS is divided into a number of fine-grained POS's. In cases where no fine-grained POS has been recognized, the fine-grained POS is the same as the coarse grained one, for example, ADJ is CPOS and its FPOS are AJP, AJCM, and AJSUP. In this research, we considered lemma as a basic feature for the joint model, and tried to improve Corbit's performance with a combination of this basic feature and Corbit default features. Corbit gets CTB file as an input file. This file includes the word's index, word, POS tag, head and dependency relation columns. First, we changed the input format to Conll. Conll format includes Coarse-grained POS tag, Fine-grained POS tag, lemma and Feats columns not found in the CTB format. The software did not exploit lemma, Feats and deprel features in the default version. Therefore, we added some new features to the software. The main research goal was POS tagging and dependency parsing on raw texts, so we chose the lemma feature, since there are lemmatizer tools for Persian and English, which can provide the required information for the software. Also, experiments and their analysis showed that some features were insufficient, and thus they were removed. Accuracy improved for both Corse-grained and Fine-grained POS tags.

- **Added features**

We tried 66 different combinations of features with lemma on Persian, and we obtained 26 features which improved accuracies. Table 1 shows the features that increased accuracies for both Coarse-grained and Fine-grained POS on Persian. Then, we tried Corbit with added features on English and accuracies, which were improved.

- **Reduced features**

We tried 66 feature combinations, and some of which reduced accuracy for both Coarse-grained POS and Fine-grained POS on Persian. It seemed these default features without lemma did not have a positive impact on accuracies. Thus, we removed these features one by one. According to the results, two Corbit default features reduced joint model accuracy for Persian; consequently, we eliminated these two features from the joint model features. Then, we tried Corbit with reduced features on English Treebank. The results showed improvement.

5. Experiments

In this research, we evaluated both the pipeline model and the joint model performance on the Persian Syntactic Dependency Treebank 1.0 and Universal Dependencies English Web Treebank v1.0. The model was trained several times, and model parameters were set. The POS tagging accuracy, dependency-parsing accuracy, and joint accuracy have been reported.

5.1. Data

We conducted experiments on the Persian Syntactic Dependency Treebank as well as Universal Dependencies English Web Treebank v1.0 for Persian and English Treebanks, respectively. Here we introduce both Treebanks briefly.

5.1.1. Persian syntactic dependency treebank

This Treebank is the first Persian dependency Treebank, and includes 29,982 sentences and 498,081 words. Its sentences have syntactic relations (based on dependency grammar) like subject, object, predicate … and POS tags like verb, noun, adjective…. Following standard practice, we adopted training, validation and test datasets. The Persian dependency Treebank was randomly split into three sets 80%, 10%, and another 10% were allocated for training, validation and test datasets, respectively. A unique feature of this Treebank is that there are 4,800 distinct verb lemmas in its sentences making it a valuable resource for educational goals [9].

5.1.2. Universal dependencies English web Treebank v1.0

Corpus consists of over 250,000 words of English weblogs, newsgroups, emails, reviews and question-answers manually annotated for syntactic structure and are designed to allow language technology researchers to develop and evaluate the robustness of parsing methods in those web domains. It contains 254,830 word-level tokens and 16,624 sentence-level tokens of webtext in 1,174 files annotated for sentence- and word-level tokenization, part-of-speech, and syntactic structure. The data is roughly evenly divided across five genres: weblogs, newsgroups, emails, reviews, and question-answers. The files were manually annotated following the sentence-level tokenization guidelines for web text and the word-level tokenization guidelines developed for English treebanks in the DARPA GALE project.

5.2. Default joint model performance

The joint model has two important parameters, beam size and iteration number. As shown in figures 3, 4, 5 and 6 increasing the iteration number and beam size improves the model for both of Persian and English. An increase in beam size of 16 to 32 and 64 significantly increases run time with little improvement in accuracy. Thus, we consider 16 for the beam size. Final results were estimated with 10 iterations, because more iterations increased run time and the accuracy improvement was not significant.

Only text from the subject line and message body of posts, articles, messages and question-answers were collected and annotated [10].

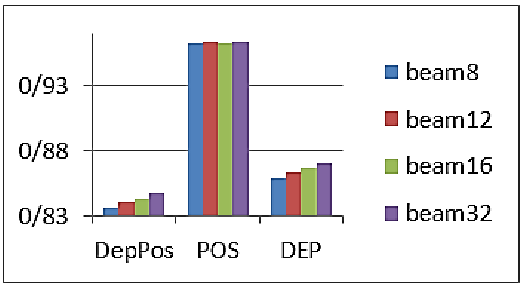

Figure 3. Study of relationship between beam size and accuracy on Persian.

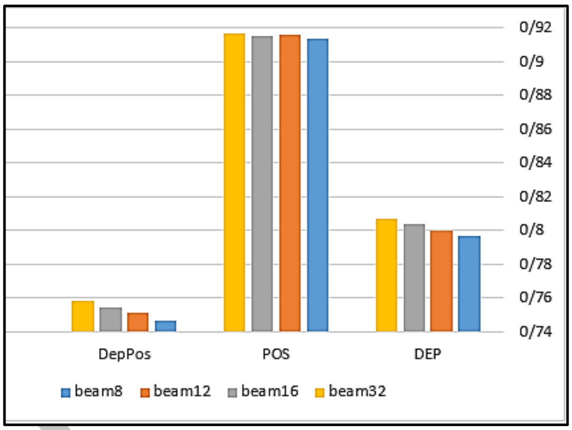

Figure 4. Study of relationship between beam size and accuracy on English.

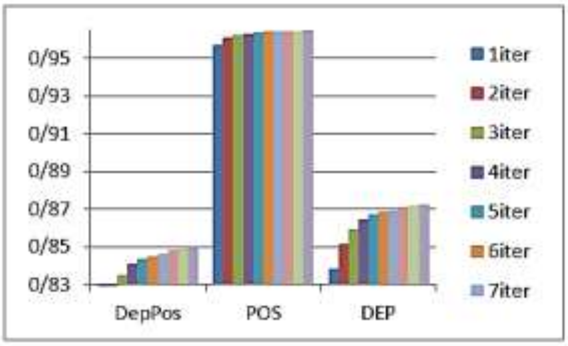

Figure 5. Study of relationship between iteration on Persian.

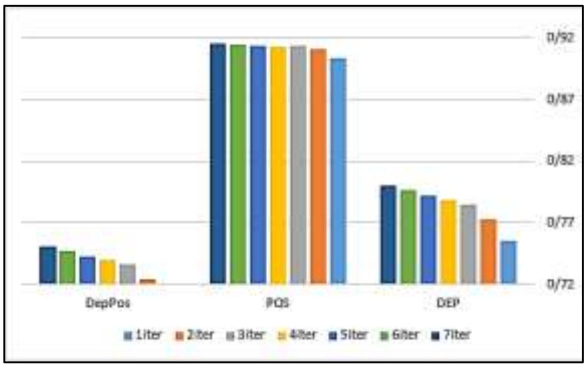

Figure 6. Study of relationship between iteration on English.

Table 1. Added Features- Lm denotes Lemma, s.w. and s.t. are the Form and Tag of the Root Word of Tree s, s.rc and s.lc are the Right- and Left-most Children of s, and ∘ Denotes the conjunction of Features.

Added Features	
1-Lm($\boldsymbol{q_0}$)	2-Lm($\mathrm{s_0}$)
3-Lm($\mathrm{s_1}$)	4-Lm($\mathrm{s_0}.rc$)
5-Lm($\mathrm{s_1}.lc$)	6-$\boldsymbol{s_0}.w \circ \mathrm{Lm}(\mathrm{s_0})$
7-$\boldsymbol{s_1}.w \circ \mathrm{Lm}(\mathrm{s_0})$	8-$\boldsymbol{s_1}.w \circ \mathrm{Lm}(\mathrm{s_1})$
9-$\boldsymbol{q}_0.w \circ \mathrm{Lm}(\boldsymbol{q_0})$	10-$\boldsymbol{s_0}.w \circ \boldsymbol{s_1}.w \circ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\mathrm{s_1})$
11-$\boldsymbol{s_0}.w \circ \boldsymbol{s_0}.t \circ \boldsymbol{s_1}.w \circ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\mathrm{s_1})$	12-$\boldsymbol{s_0}.t \circ \boldsymbol{s_1}.w \circ \boldsymbol{s_1}.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\mathrm{s_1})$
13-$\boldsymbol{s_0}.w \circ \boldsymbol{s_0}.t \circ \boldsymbol{s_1}.w \circ \boldsymbol{s_1}.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\mathrm{s_1})$	14-$\boldsymbol{s_0}.t \circ \boldsymbol{q_0}.t \circ\ \mathrm{Lm}(\mathrm{s_0})$
15-$\boldsymbol{s_0}.t \circ \boldsymbol{s_1}.t \circ \boldsymbol{q_0}.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\mathrm{s_1})$	16-1$\boldsymbol{s_0}.w \circ \boldsymbol{s_1}.t \circ \boldsymbol{q_0}.t \circ\ \mathrm{Lm}(\mathrm{s_0})$
17-$\boldsymbol{s_0}.t \circ \boldsymbol{q_0}.t \circ \boldsymbol{q_1}.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\mathrm{q_0})$	18-$\boldsymbol{s_0}.t \circ \boldsymbol{s_1}.t \circ \boldsymbol{s_1}.lc.t \circ\ \mathrm{Lm}(\mathrm{s_0})$
19-$\boldsymbol{s_0}.t \circ \boldsymbol{s_1}.t \circ \boldsymbol{s_1}.rc.t \circ\ \mathrm{Lm}(\mathrm{s_0})$	20-$\boldsymbol{s_0}.t \circ \boldsymbol{s}_0.rc.t \circ \boldsymbol{s_1}.t \circ\ \mathrm{Lm}(\mathrm{s_0})$
21-$\boldsymbol{s_0}.t \circ \boldsymbol{s_1}.t \circ \boldsymbol{s_1}.lc.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\boldsymbol{s}_1) \circ \mathrm{m}(\boldsymbol{s}_1.lc)$	22-$\boldsymbol{s_0}.t \circ \boldsymbol{s_0}.lc.t \circ \boldsymbol{s_1}.t \circ\ \mathrm{Lm}(\mathrm{s_0})$
23-$\boldsymbol{s_0}.t \circ \boldsymbol{s_0}.rc.t \circ \boldsymbol{s_1}.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\mathrm{s_0}.\mathrm{rc}) \circ\ \mathrm{Lm}(\mathrm{s_1})$	24-$\boldsymbol{s_0}.w \circ \boldsymbol{s_1}.t \circ \boldsymbol{s}_1.rc.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ\ \mathrm{Lm}(\mathrm{s_1}) \circ\ \mathrm{Lm}(\mathrm{s_1}.\mathrm{rc})$
25-$\boldsymbol{s_0}.w \circ \boldsymbol{s_1}.t \circ \boldsymbol{s}_0.lc.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ\ \mathrm{Lm}(\mathrm{s_1}) \circ\ \mathrm{Lm}(\mathrm{s_0}.\mathrm{lc})$	26-$\boldsymbol{s_0}.t \circ \boldsymbol{s_1}.t \circ \boldsymbol{s_2}.t \circ\ \mathrm{Lm}(\mathrm{s_0}) \circ \mathrm{Lm}(\mathrm{s_1})$

Table 2. Reduced features.

Reduced Features	
1) $\boldsymbol{s_0}.t \circ \boldsymbol{s_1}.t \circ \boldsymbol{s_2}.t$	2) $\boldsymbol{s_0}.w \circ \boldsymbol{s_0}.t \circ \boldsymbol{s_1}.t$

Table 3. Pipeline model results.

Pipeline Model	Lang.	Tag Acc	DepPos
CPOS	Persian	0.9742	0.766494
	English	0.9468	0.747097
FPOS	Persian	0.9611	0.860225
	English	0.9458	0.734691

5.3. Pipeline model performance

Data has been tagged with the Stanford tagger, and then the tagged data was parsed with a baseline dependency parser. In the pipeline method, Corbit software just does the dependency parsing task using gold POS tags. POS tagging and dependency parsing for Coarse-grained and Fine-grained POS tags are shown in table 3.

Table 4 shows that the default joint model has a better performance than the pipeline model for Coarse-grained POS (8% and 0.73% improvement for Persian and English, respectively).

Table 5 shows that the accuracy of the joint model for Fine-grained POS on Persian was 2.4% less than the pipeline model. For English, the results of Joint model and pipeline model are almost equal.

Table 4. Model First Result for CPOS.

Model	Lang.	DepPos	POS	DEP
Joint model	Persian	0.849489	0.964 459	0.872165
Baseline	Persian	0.766494	0.974 2	0.766494
Joint model	English	0.754374	0.915 063	0.803961
Baseline	English	0.747097	0.911 404	0.797360

Table 5. Model first result for FPOS.

model	Language	DepPos	POS	DEP
Joint model	Persian	0.836072	0.949871	0.872363
Baseline	Persian	0.860225	0.9611	0.860225
Joint model	English	0.735049	0.901781	0.789367
Baseline	English	0.734691	0.899833	0.791117

5.4. Joint model performance with gold POS tag

We ran Corbit software with the gold POS tag on dependency parsing mode, and the best dependency result obtained for the default joint model is shown in table 6.

Table 6. Dependency parsing results with gold.

model	Lang.	DepPos
CPOS	Persian	0.893478
	English	0.764374
FPOS	Persian	0.896646
	English	0.755049

- **Improved joint model performance**

The effect of each added feature on Corbit software accuracy for Coarse-grained and Fine-grained POS is shown in table 7. A positive number means an increase and negative number means a decrease in accuracy. We tried to choose features that improved both Coarse-grained and Fine-grained POS accuracies. The other features that significantly reduced either CPOS or FPOS or both accuracies were not included in the features list. A 0.03% increase in accuracy for both CPOS and FPOS meant improvement, but in the case where only one of the POS increased and the other decreased, the feature was considered useful only if the sum of the increase and decrease was more than 0.04%; otherwise, this feature added significantly to the run time. As mentioned in section 5.1, an increase in iteration number increased accuracy. Results showed that a gradient shift of accuracy in the 1st to 5th iterations was more than in the 6th to 10th iterations. Therefore, we used 5th iteration results. It is clear that if features show improvement in 5 iterations, they have improvement with fewer gradients in 6 to 10 iterations. Added features are listed in table 1. In each step, we add one feature to the other features, and measure the changes in accuracy for both CPOS and FPOS.

Default joint model accuracy (*), joint model accuracy after adding features (**) and reducing features (***) with 5 iterations and a beam size of 16 on the validation dataset is shown in table 8. Joint model accuracy by adding features on Persian has improved 0.7% for CPOS and 0.8% for FPOS. After reducing 2 default features, the joint model accuracy increased 0.3% for CPOS and 0.2% for FPOS on Persian.

Totally, For Persian, the joint model accuracy increased 1% for CPOS and 1% for FPOS. Corbit's accuracy with added features had 0.29% improvement for CPOS and 33% improvement for FPOS on English. The joint model accuracy with reduced features improved CPOS and FPOS 0.24% and 0.31%, respectively. Therefore, Corbit's accuracy for English improved 0.53% and 0.64% for CPOS and FPOS in order. As we mentioned in section 5.1, the Persian dependency Treebank includes 29,982 sentences and 498,081 words but Universal Dependencies English Web Treebank contains 254,830 word-level tokens and 16,624 sentence-level tokens of web texts. It shows that Persian Treebank's sentences and words are almost twice, so we achieved higher improvement for Persian comparing to English.

The default joint model and joint model accuracy after optimization on test data is shown in table 9. As can be seen, the joint model accuracy improved 0.83% for CPOS and 0.49% for FPOS on Persian and 0.40% for CPOS and 0.53% for FPOS on English after feature optimization. The DEP for CPOS increased 0.81% and 0.37% for Persian and English, respectively. The FPOS improved 0.4% for Persian and 0.53% for English.

Table 7. Added features effect on increase and decrease of accuracy for Persian.

#	CPOS	FPOS	#	CPOS	FPOS
1	0.2	0.4	2	0.2	-0.03
3	0.04	0.08	4	-0.02	0.16
5	0.27	0	6	-0.02	0.08
7	0.12	0.08	8	-0.05	0.11
9	-0.08	0.28	10	0.18	0.16
11	0.16	0.34	12	0.03	0.01
13	-0.01	0.06	14	0.11	0.19
15	0.22	0.14	16	-0.04	0.13
17	0.01	0.21	18	0.11	0.09
19	0.03	0.26	20	0.28	0.26
21	0.13	0.13	22	0.06	0.12
23	0.13	0.14	24	0.07	-0.01
25	0.3	0.04	26	0.05	0.4

6. Conclusion

This research is based on a joint POS tagging and dependency-parsing model used on the Chinese language. POS tag and dependency relationship are language-specific features called morphological features [11]. First, we reconciled the model with Persian and English. In experiments, the default joint model had improvement over the pipeline model for CPOS. Then, we considered lemma as a key feature for feature optimization on Persian and English. We studied different combinations of lemma with default features. The combinations that had a subtractive effect were removed.

Finally, a 1% improvement for Persian and almost 0.5% for English was obtained for CPOS and FPOS.

In this research, we focused on Persian and English but adding lemma is possible for other languages and the improved joint model is language-independent.

Table 8. Joint model accuracy on validation dataset with 5 iterations, before adding features of table 3, after adding features of table 3, after adding features of table 3 and reducing features of table 4.

POS	Lang.		DepPos	POS	DEP
CPOS	Persian	*	0.843418	0.963179	0.867129
		**	0.850632	0.96554	0.872671
		***	0.853404	0.96624	0.875069
	English	*	0.754374	0.915063	0.803961
		**	0.757220	0.914506	0.806285
		***	0.759609	0.914864	0.808465
FPOS	Persian	*	0.829847	0.949251	0.866359
		**	0.837963	0.952532	0.872561
		***	0.839965	0.952913	0.874497
	English	*	0.735049	0.901781	0.789367
		**	0.738349	0.901543	0.794457
		***	0.741441	0.902100	0.797610

Table 9. Joint model accuracy on test dataset with 10 iterations before and after features optimization.

POS	Lang.		DepPos	POS	DEP
CPOS	Persian	*	0.847674	0.964827	0.868994
		**	0.855900	0.966992	0.877040
	English	*	0.756136	0.917278	0.802518
		**	0.760179	0.917085	0.806239
FPOS	Persian	*	0.837532	0.950018	0.873023
		**	0.842490	0.953348	0.877034
	English	*	0.738126	0.904686	0.790883
		**	0.743433	0.903812	0.795509

Acknowledgment

This study has been supported by Iran University of Science and Technology of Iran. We gratefully acknowledge the support of the research department.

References

[1] Hatori, J. Matsuzaki, T., Miyao, Y. & Tsujii, J. I. (2012). Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 1045-1053.

[2] Bohnet, B. & Nivre, J. (2012). A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1455-1465.

[3] Hatori, J., Matsuzaki, T., Miyao, Y. & Jun'ichiTsujii. (2011). Incremental Joint POS Tagging and Dependency Parsing in Chinese. In IJCNLP, pp. 1216-1224.

[4] Li, Z., Zhang, M., Che, W., Liu, T., Chen, W. & Li, H. (2011). Joint models for Chinese POS tagging and dependency parsing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1180-1191.

[5] Li, Z., Zhang, M., Che, W., Liu, T. & Chen, W. (2012). A Separately Passive-Aggressive Training Algorithm for Joint POS Tagging and Dependency Parsing. In COLING, pp. 1681-1698.

[6] Seraji, M., Megyesi, B. & Nivre, J. (2012). Dependency parsers for Persian. In COLING, pp. 35-44.

[7] Khallash, M., Hadian, A., & Minaei-Bidgoli, B. (2013). An Empirical Study on the Effect of Morphological and Lexical Features in Persian Dependency Parsing. In Fourth Workshop on Statistical Parsing of Morphologically Rich Languages, p. 97-107.

[8] Toutanova, K., Klein, D., Manning, C. D. & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 173-180.

[9] Rasooli, M. S., Kouhestani, M. & Moloodi, A. (2013). Development of a Persian syntactic dependency treebank. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies, pp. 306-314.

[10] Bies, A., Mott, J., Warner, C. & Kulick, S. (2012). English Web Treebank. Technical Report LDC2012T13, Linguistic Data Consortium, Philadelphia, PA.

[11] Zahedi, M. & Arjomandzadeh, A. (2015). A new model for persian multi-part words edition based on statistical machine translation. Journal of AI and Data Mining.

نشریه هوش مصنوعی و داده کاوی

دانشگاه شاهرود

ارائه و بهسازی مدل توأم برچسب‌زنی اجزای سخن و تجزیه‌ی وابستگی

عاطفه پاکزاد و بهروز مینایی بیدگلی*

دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.

ارسال ۲۰۱۵/۰۹/۲۵؛ پذیرش ۲۰۱۵/۱۰/۱۲

چکیده:

تجزیه‌ی وابستگی راهی برای تجزیه‌ی نحوی زبان طبیعی است که به صورت خودکار به تجزیه و تحلیل ساختار وابستگی جملات پرداخته و برای هر جمله‌ی ورودی یک گراف وابستگی ایجاد می‌کند. برچسب‌زنی اجزای سخن برای انجام تجزیه‌ی وابستگی یک پیش‌نیاز است. عموما تجزیه‌گرهای وابستگی به صورت مرحله‌ای پیوسته، وظایف برچسب‌زنی و تجزیه‌ی وابستگی را به صورت دو گام متوالی انجام می‌دهند. در این مدل‌ها خطای ناشی از برچسب‌زنی در تجزیه‌ی وابستگی انتشار می‌یابد، همچنین در حین برچسب‌زنی از اطلاعات مفید نحوی استفاده نمی‌کند. هدف از ارائه‌ی روش‌های توأم، کاهش همزمان خطای هر دو وظیفه‌ی برچسب‌زنی اجزای سخن و تجزیه‌ی وابستگی است. در این پژوهش مدل توأم بر روی زبان فارسی و انگلیسی با استفاده از نرم‌افزار Corbit مورد آزمایش قرار گرفته و ویژگی‌های مدل بهینه‌سازی شده است که سبب بهبود در دقت مدل توأم گردیده است. نرم‌افزار Corbit پیاده‌سازی یک روش توأم مبتنی بر گذار برای وظایف تقسیم‌بندی کلمه، برچسب‌زنی اجزای سخن و تجزیه‌ی وابستگی است. در این پژوهش دقت توأم برچسب‌زنی اجزای سخن و اتصال بدون برچسب تجزیه‌ی وابستگی بر روی داده‌های آزمون در زبان فارسی برای برچسب‌های درشت برابر ۸۵,۵۹ درصد و برای برچسب‌های ریز ۸۴,۲۴ درصد به‌دست آمده است. همچنین بر روی زبان انگلیسی ما به دقت ۷۶,۰۱ برای برچسب های درشت و ۷۴,۳۱ برای برچسب های ریز دست یافته‌ایم.

کلمات کلیدی: مدل توأم، اجزای سخن، تجزیه‌ی وابستگی، زبان فارسی.